

AUTOMATION IN PYTHON USING SPEECH RECOGNITION

Mrs.P.Dhakshayeni¹, S. Rithika Irene², V. Thanga Ishwarya³, M. Vaishnavi⁴

¹Associate Professor, Department of Information Technology, Jeppiaar SRR Engineering College, Padur Chennai, Tamil Nadu, India

^{2,3,4}B.Tech, Department of Information Technology, Jeppiaar SRR Engineering College, Padur Chennai, Tamil Nadu, India

Abstract - Automation in python using speech recognition, helps to compile and execute commands using a voice assistant written in python code. The event of compiling and executing commands is done by speech command. The basic need is to have the database containing events of creation, detection, modified recorded to avoid ambiguity and to provide correct data of currently available python scripts and java codes in the specified directory. The directory specified acts as a root directory and it notifies all events of the system from that mentioned directory as the starting point to monitor. The speech assistant invokes the creation of batch file for the file specified through voice. Which will in turn invoke the python script needs to be specified.

Key Words: Python, Command prompt, Batch file, Directory, Database.

1. INTRODUCTION

The main aim of the project is to automatically open the command prompt and do auto compilation and execution of the codes with voice command. The voice command involves the name and type of the file which is stored in the directory specified to the watchdog. The major concept is to create batch file for the script to be executed through voice command. The outline of this project is the invoking of the sub-process module to run batch scripts which in return execute a file that needs to be compiled and executed. Thus a single batch file is created each time a program is asked to execute. The code to run the python scripts through batch file or java code through batch file is written on the file created. And the process to invoke cmd prompt is run to get the output of the code that was asked to execute through voice command. The batch file destroys itself after the execution of the code. Here the files managed by the watchdog is only accessed.

In this paper second section consists of relative work, third section consists of proposed system, fourth section consists of UML diagrams, fifth section consists of modules, sixth section consists of conclusion, seventh section consists of future enhancement and eighth section consists of references.

2. RELATIVE WORKS

Advancements in technologies and new project ideas evolve every year exponentially. This is necessary to establish a system that overcomes the existing drawbacks

and continually grow over time. In order to obtain this, many enthusiasts invest their priced energy and time to create new ideas that function better than the living system. Research papers play critical job in the demonstration of these ideas. Studies suggest that majority of research time is spent on drafting the paper. Drafting of paper doesn't require human intelligence and can be piled upon automated machines. Many Technology innovations like speech to text, machine learning and big data have already reduced the heavy load from user shoulders up to a great degree. A good combination of these libraries can be employed to develop automated paper bots. Any research includes analysis of Idea and paper drafting. Former task requires human Intelligence and hence is highly difficult for machines than humans the latter task requires labour work based on set of rules and hence is easier and faster for machines than it is for humans.

3. PROPOSED SYSTEM:

With just the file name said to the computer, it opens the command prompt and helps in execution by converting speech to text.

4. UML DIAGRAMS:

USECASE DIAGRAM: A use case diagram is one of the set of scenarios that describing an interaction between a user and a system. A use case diagram displays a relationship among actors and use cases. Two main components are a user or another system that will interact with the system modeled. A use case diagram is one of an external view of the system that represents action the user may perform in order to complete a task.

SEQUENCE DIAGRAM: A kind of interaction diagram that shows how procedure manage with one another and in what order is called Sequence diagram. Build of message sequence diagrams are sometimes called event diagrams, event scenarios and timing diagram.

COLLABORATION DIAGRAM: A collaboration diagrams is a group of UML diagrams called interaction diagrams. Collaboration diagrams are like sequence diagrams, show how objects interact over the course of time. However, instead of showing the sequence by numbering the messages of the diagram. This makes easier to show how the objects are linked together, but harder to see the sequence at a glance.

5. MODULES:

1. Creation and setting the path to monitor.
2. watchdog to look for files system changes
3. Updating database.
4. Voice input to get the file name.
5. Creating batch file to execute.

i. CREATION AND SETTING THE PATH TO MONITOR

The very first step of the project is to get the root directory from the user from where the file system needs to be monitored, so that any event which occurs within that directory can be analysed and a proper data with no redundant files or directories can be maintained. And this step also involves selecting of the languages which needs to be monitored in that file system namely python and java languages with corresponding execution paths to be chosen as there can be multiple python or java installed of various versions. The path will be monitored from the time the system is on and till the system gets off. Making sure all events are gathered to know the files and sub-directories of the root directory created from user inputs.

ii. WATCH DOG TO LOOK FOR FILE SYSTEM CHANGES

As the name indicates, watchdog is created at the moment the path is set. Hence forth this watchdog notifies any even that takes place from that root directory till the last subdirectory created at the instance of time. For example events like creation of file or sub-directory, deletion of file or sub-directory, renaming any file or sub-directory belonging to that root-directory.

iii. UPDATING DATABASE

Events like creation of file or sub-directory, deletion of file or sub-directory, renaming any file or sub-directory belonging to that root-directory should be monitored to know which files exists, which files are deleted, to get the new name of the file renamed and also to have unique name for each file. If the same file name happens to occur in future it pops up a windows message to indicate that file name already exists and to follow uniqueness in the file name.

iv. VOICE INPUT TO GET THE FILE NAME

The microphone of the windows system is used to get the input from the user. Which is converted to text and compared with the files names stored in the database. If exists the file is further processed.

v. CREATING BATCH FILE TO EXECUTE

The python file creates a batch script which in turn invokes the python script which needs to be executed by for the user. Thus it is a python script – batch script – python script

process of invocation. The batch file will destroy itself after a point of time. It will invoke the command prompt and it will itself write the code for compiling and executing the code.

6. CONCLUSION

Thus this will reduce the human effort of running the java or python codes to compile or execute by typing. The file name when said is invoking the programs written. It will in turn show the windows command prompt and shows error if any or shows the output.

7. FUTURE ENHANCEMENT

1. It can be implements for other operating systems
2. As the future network growth is high the executing will become even more faster.
3. As the speed of showing the output depends on the speed of internet.
4. Can be implemented for various languages also.
5. It can be implemented for various language editors

8. REFERENCES

1. <https://www.geeksforgeeks.org/personal-voice-assistant-in-python/>
2. <https://blog.miguelgrinberg.com/post/how-to-make-python-wait>
3. <https://realpython.com/python-speech-recognition/>
4. <https://www.devdungeon.com/content/pyinstaller-tutorial>
5. https://www.google.co.in/search?biw=1708&bih=819&sxsrf=ALeKk00eKm--yvwcc9oK-H6tiUt-h4PZYA%3A1583485386781&ei=yhFiXsqgL5KR4-EP5NG-2AU&q=creating+window+icon+to+run+a+a+application+using+python&oq=creating+window+icon+to+run+a+a+application+using+python&gs_l=psy-ab.3...24803.41845..42665...12.3..0.219.3447.18j14j1.....0...1..gws-wiz.....0i71j35i304i39j33i10.NWJgVoXJEq&ved=0ahUKEwjKrfHfvoXoAhWSyDgGHeSoD1sQ4dUDCAo&uact=5#kpvalbx=_GRJiXsD3Hvef4-EPibik-AY27
6. <http://www.blog.pythonlibrary.org/2010/01/23/using-python-to-create-shortcuts/>
7. <https://www.geeksforgeeks.org/autorun-a-python-script-on-windows-startup/>
8. <https://superuser.com/questions/395015/how-to-open-the-recycle-bin-from-the-windows-command-line/395140>
9. <https://stackoverflow.com/questions/42089399/how-can-i-delete-files-from-recycle-bin-in-windows-python>

10. <https://stackoverflow.com/questions/11351043/enclose-a-variable-in-single-quotes-in-python>
11. <https://medium.com/@ageitgey/python-3-quick-tip-the-easy-way-to-deal-with-file-paths-on-windows-mac-and-linux-11a072b58d5f>